

A HIGHER STRUCTURE IDENTITY PRINCIPLE

DIMITRIS TSEMENTZIS

ABSTRACT. We prove a Structure Identity Principle for theories defined on types of h -level 3 by defining a general notion of saturation for a large class of structures definable in the Univalent Foundations.

Formalizing mathematics in the framework of the Univalent Foundations [Uni13] presents unique challenges and opportunities. One of the main opportunities is the ability to formalize higher-level mathematics (categories, higher categories etc.) in an invariant way by imposing appropriate saturation conditions. This is done, for example, in [Uni13] for category theory (cf. Definition 9.1.6). An interesting challenge is how to express general saturation conditions that apply to wide classes of structures that can be formalized in UF. The main contribution of this paper is to provide such a general definition of a saturation condition for a wide class of definable structures and to use it to prove a Structure Identity Principle for all “category-level” (or “3-level”) such structures.

To illustrate, consider the following theorem from [Uni13]:

Theorem 0.1 ([Uni13], Theorem 9.4.16). *For any univalent categories \mathcal{C} and \mathcal{D} , the type of categorical equivalences $\mathcal{C} \simeq_{\text{cat}} \mathcal{D}$ is equivalent to $\mathcal{C} =_{\text{UniCat}} \mathcal{D}$.*

By regarding univalent categories as a “saturated” version of an unsaturated structure (i.e. of precategories) we can regard the above result as a specific instance of a more general result of the following form:

Pre-Theorem 0.2. *For any saturated models \mathcal{M} and \mathcal{N} of an \mathcal{L} -theory \mathbb{T} , the type of \mathcal{L} -equivalences $\mathcal{M} \simeq_{\mathcal{L}} \mathcal{N}$ is equivalent to $\mathcal{M} =_{\text{SatMod}_{\mathbb{T}}} \mathcal{N}$.*

The purpose of this paper is to make precise and prove a result of this form, such that Theorem 9.4.16 follows as a special case (as an assurance of adequacy).

This is done as follows. An “ \mathcal{L} -theory \mathbb{T} ” will be given by a theory over a FOLDS signature \mathcal{L} in the sense of Makkai [Mak95], i.e. a finite inverse category \mathcal{L} . A “model \mathcal{M} ” of \mathbb{T} (or just simply an “ \mathcal{L} -structure”) will be given by an interpretation of FOLDS into HoTT as has been described in [Tse16]. An “ \mathcal{L} -equivalence” between two such models \mathcal{M} and \mathcal{N} will be given by Makkai’s notion of FOLDS \mathcal{L} -equivalence, which in set theory is defined as the existence of a span of fiberwise surjective \mathcal{L} -homomorphisms from \mathcal{M} to \mathcal{N} . What remains is to define what a “saturated” model is in this setting and indeed this is the main contribution of this paper.

Date: February 28, 2017.

2010 Mathematics Subject Classification. 03G99, 03B15, 03B22, 03C99.

Key words and phrases. Univalent Foundations, Categorical Logic, Homotopy Type Theory.

The general definition of saturation is based on the following intuitive idea: a structure is saturated if any two indistinguishable components in it are identical. To make this idea precise we will define for any FOLDS signature \mathcal{L} a general notion of isomorphism for any sort K in \mathcal{L} and any $x, y: K$ in the form of an \mathcal{L} -formula $x \cong y$. The intuition behind $x \cong y$ is that “ x and y are indistinguishable by any sort which depends on K , in any position”. Spelling out exactly how “indistinguishable” and “position” are to be formalized constitutes the main task of the syntactic portion of the paper. Roughly, this is done by a mutually inductive definition of a formula $\text{Ind}(x, y)$ expressing indistinguishability (up to equivalence) with respect to sorts “upstairs” (Definition 2.5) and then a notion of equivalence which says that the indistinguishability relation $\text{Ind}(x, y)$ is a bijection (up to indistinguishability) (Definition 2.10).

We can then interpret the formula $x \cong y$ for any \mathcal{L} -structure \mathcal{M} and given this we can define what it means for such an \mathcal{M} to be saturated at any sort K and also associated notions of total saturation, saturation at a given level etc. This forms the key definition of this paper (Definition 3.2) which provides a very wide notion of saturation of which the known examples of saturated structures become special instances. For example, we will show that the totally saturated models of the theory of categories are the univalent categories (Proposition 4.13) whereas precategories are the “2-saturated” such models (Proposition 4.10).

Finally, given this general notion of saturation, we will prove a precise version of Pre-Theorem 0.2 for FOLDS signatures \mathcal{L} of “height” 3. Signatures of this height can be thought of as expressing theories whose saturated models are given by structures on 1-types, i.e. on types of h -level 3. The theory of precategories is an example of such a theory. This is done in Theorem 4.9, the main result of this paper, which can be thought of as a Structure Identity Principle for structures of h -level 3.

Outline of the Paper. In Section 1 we fix notation and introduce the relevant concepts from FOLDS. In Section 2 we give the definition of $x \cong y$ for any FOLDS signature \mathcal{L} and any $x, y: K$ (Definitions 2.5 and 2.10). In Section 3 we define saturation for FOLDS \mathcal{L} -structures in type theory (Definition 3.2) and prove certain useful properties of our definitions. Finally, in Section 4 we internalize the notion of FOLDS equivalence in type theory and use this definition to prove a Structure Identity Principle for 3-level structures (Theorem 4.9). We also provide proofs that our higher Structure Identity Principle gives the right answer in the case of the theory of categories (Propositions 4.10 and 4.13).

1. PRELIMINARIES AND NOTATION

1.1. FOLDS. By FOLDS we understand the system of Makkai introduced in [Mak95]. We here introduce the relevant concepts. A *FOLDS signature* \mathcal{L} is a finite inverse category. We assume each signature \mathcal{L} comes with a *level* function $l: \text{ob}\mathcal{L} \rightarrow \mathbb{N}$ defined

as follows

$$l(K) = \begin{cases} 1 & \text{if } K \text{ is the codomain only of } 1_K \\ \sup_{f: K_f \rightarrow K} l(K_f) + 1 & \text{otherwise} \end{cases}$$

Note that “top-level” sorts are assigned level 1, contrary to Makkai’s original definition. This is done to foreshadow that these sorts, when saturated, will be interpreted as types of h -level 1. However, we will write $K > K'$ to denote that K has a *lower* level than K' (i.e. that $l(K) < l(K')$ in order to think of $K > K'$ as saying that K is “higher” (and therefore depends on) K'). We also write $h(\mathcal{L})$ for the *height* of \mathcal{L} , i.e. the maximum level of any sort K in \mathcal{L} (since we are assuming that \mathcal{L} is finite, the height will also always be finite). For any $K \in \text{ob}\mathcal{L}$ we will write $\mathcal{L} \downarrow K$ for the comma category under K in \mathcal{L} . We will generally use the notation $\mathcal{L}(K, K')$ for the hom-sets of \mathcal{L} .

Variables for \mathcal{L} are given by a functor $V: \mathcal{L} \rightarrow \mathbf{Set}$ satisfying certain obvious conditions, e.g. that for any $K \neq K'$, $V(K)$ and $V(K')$ are countably infinite disjoint sets etc. We will write $x: K$ for $x \in V(K)$, for any $K \in \text{ob}\mathcal{L}$. For any $x: K$ and $f \in \mathcal{L}(K, K_f)$ we write x_f for $V(f)(x)$. It is helpful to think of f as a *position* and x_f as the variable that x depends on in position f . We write $\text{dep}(x)$ for the set $\{x_f \mid \text{dom}(f) = K\}$, i.e. the set of *dependent variables* of x and ∂x for the the set of dependent variables of x but omitting x , i.e the *boundary* of x .

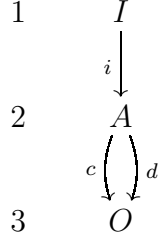
A *context* is a finite subfunctor of V . A *context morphism* is a natural transformation $s: \Gamma \rightarrow \Delta$. For any two contexts Γ and Δ we write $\Gamma \cup \Delta$ for their union as subobjects of V , i.e. the functor that takes $K \mapsto \Gamma(K) \cup \Delta(K)$. It is not hard to see that both $\text{dep}(x)$ and ∂x define contexts. We will use the symbol $=$ for equality of contexts and variables.

We define the syntax associated to \mathcal{L} by an inductive definition in the usual way (except the clauses for quantifiers are a little more involved than first-order logic) and we write $\mathbf{Form}_{\mathcal{L}}(\Gamma)$ for the set of \mathcal{L} -formulas in context Γ . We write $\text{FV}(\phi)$ for the free variables of an \mathcal{L} -formula ϕ . Substitution along a context morphism is defined in the expected way and we write $s(\phi)$ for the result of substituting the variables in a formula ϕ in context Γ along $s: \Gamma \rightarrow \Delta$. As usual, we consider formulas up to α -equivalence, denoted by \equiv_{α} , i.e. up to renaming of their bound variables. A *theory* \mathbb{T} in \mathcal{L} is a set of \mathcal{L} -sentences. It is also useful (for the sake of Definition 2.5) to define the coarser notion of *contextual equivalence* of formulas ϕ (in context Γ) and ψ (in context Δ), denoted \sim_c , as follows: $\phi \sim_c \psi$ iff there exists an isomorphism $s: \Gamma \rightarrow \Delta$ such that $s(\phi) \equiv_{\alpha} \psi$. In other words, two formulas are contextually equivalent if we can rename the free variables of one in such a way as to make it α -equivalent to the other. This notion is helpful in distinguishing repeat uses of the same variable in formulas, e.g. $\phi(x, x)$ is not contextually equivalent to $\phi(x, y)$ but $\phi(x, y)$ is contextually equivalent to $\phi(z, w)$.

For $R \in \text{ob}\mathcal{L}$ with $l(R) = 1$ and $\alpha: R$ we write $R(\alpha)$ as syntactic sugar for the formula $\exists \alpha: R. \top$. This means that $\text{FV}(R(\alpha)) = \partial \alpha$ which means that if $\partial \alpha = \partial \beta$ then $R(\alpha) \equiv_{\alpha} R(\beta)$.

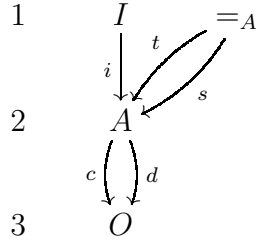
For a set of variables Γ (not necessarily a context) we write $\forall\Gamma\phi$ for the universal closure of ϕ under all variables in Γ . When this is done we will assume that $\forall\Gamma\phi$ is well-formed, i.e. that $\text{FV}(\forall\Gamma\phi)$ is a well-formed context. For a given set of formulas S , by $\bigwedge S$ we will mean the conjunction of all formulas in S (and similarly for $S_1 \wedge S_2$). We will also reserve the symbol \Leftrightarrow for logical equivalence over a standard deductive system for FOLDS (cf. [Mak95] for details). Alternatively, for the purposes of this paper one may take the logical consequence relation \Rightarrow to mean simply derivability in the type-theoretic semantics of FOLDS formulas (cf. [Tse16] for a detailed explanation of these semantics). For example without any loss of content one may simply take something like $\phi \Rightarrow \psi$ to mean that given a term of the interpretation of the formula ϕ as a type one can produce a term of the type ψ in type theory. We will write $\mathbb{T} \models \phi \Leftrightarrow \psi$ to denote \mathbb{T} -provable logical equivalence of ϕ and ψ .

Several examples of FOLDS signatures will come in handy for the purpose of illustration, so we define them here, with the level of each sort displayed on the left. We will denote by \mathcal{L}_{rg} the FOLDS signature



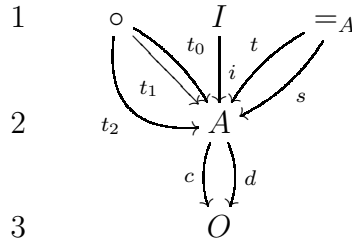
subject to the relation $di = ci$. \mathcal{L}_{rg} can be thought of as a signature useful for formalizing reflexive graphs.

By $\mathcal{L}_{\text{rg}}^{\text{=A}}$ we will denote the FOLDS signature extending \mathcal{L}_{rg} where we add an “equality” on A



subject to the relations $di = ci, ds = dt, cs = ct$.

By \mathcal{L}_{cat} we will denote the following FOLDS signature



subject to the same relations as $\mathcal{L}_{\text{rg}}^{\text{=A}}$ in addition to the following:

$$dt_0 = dt_2, ct_1 = ct_2, dt_1 = ct_0$$

$$ds = dt, cs = ct$$

By \mathbb{T}_{cat} we will denote the \mathcal{L}_{cat} -theory of categories axiomatized in the usual way, i.e. by adding axioms expressing that \circ is an associative operation for which I picks out a right and left unit etc. (cf. [Tse16] for details).

1.2. Homotopy Type Theory. On the semantic side, we will assume that we will work with some Homotopy Type Theory to which we will refer generically as *type theory*. The system in [Uni13] will do just fine, for example. In particular, we will assume that all the types in terms of which \mathcal{L} -structures are defined live in some univalent universe \mathcal{U} . The terms *proposition*, *set*, *n-type*, *h-level*, *precategory*, *univalent category* etc. refer to the standard definitions as given e.g. in [Uni13]. We will denote by **PreCat** the type of precategories and by **UniCat** the type of univalent categories. For a given type A and $a, b : A$ we will write $a =_A b$ for the identity type of a and b , often abbreviating to $a = b$ when A is clear from the context. For a given $f : A \rightarrow B$ we will also use the abbreviation $\text{bij}(f)$ for the type

$$\left(\prod_{x, x' : A} f(x) = f(x') \rightarrow x = x' \right) \times \left(\prod_{y : B} \sum_{x : A} f(x) = y \right)$$

We use the notation $\|A\|_n$ for the n -truncation of a type A , where n will denote the h -level, e.g. for $n = 1$, $\|A\|_1$ is the propositional truncation.

Any FOLDS signature \mathcal{L} can be translated as data in type theory in a straightforward way, giving us a notion of a type **Str $_{\mathcal{L}}$** of \mathcal{L} -structures a term \mathcal{M} of which can be thought of as an \mathcal{L} -structure (cf. [Tse16] for details). This is done by extracting a Σ -type from \mathcal{L} by induction on the level of \mathcal{L} . For example, we have

$$\mathbf{Struc}_{\mathcal{L}_{\text{rg}}} =_{\text{df}} \sum_{O : \mathcal{U}} \sum_{A : O \rightarrow O \rightarrow \mathcal{U}} \prod_{x : O} A(x, x) \rightarrow \mathcal{U}$$

Whenever we pick a specific \mathcal{L} -structure $\mathcal{M} : \mathbf{Struc}_{\mathcal{L}}$ we will denote by $K^{\mathcal{M}}$ the interpretation of any sort $K \in \text{ob}\mathcal{L}$. For example, the data of $\mathcal{M} : \mathbf{Struc}_{\mathcal{L}_{\text{rg}}}$ can be written as $\langle O^{\mathcal{M}}, A^{\mathcal{M}}, I^{\mathcal{M}} \rangle$. To further illustrate the notation, we can for instance define

$$\mathbf{Struc}_{\mathcal{L}_{\text{rg}}^A} =_{\text{df}} \sum_{\mathcal{M} : \mathbf{Struc}_{\mathcal{L}_{\text{rg}}}} \left(\prod_{x, y : O^{\mathcal{M}}} A^{\mathcal{M}}(x, y) \times A^{\mathcal{M}}(x, y) \rightarrow \mathcal{U} \right)$$

Similarly, all \mathcal{L} -formulas can be translated under the usual propositions-as-types translation of the quantifiers and logical connectives (with \exists interpreted as truncated Σ and \vee interpreted as truncated $+$ ensuring that the interpretations are all h -props). For example, for the \mathcal{L}_{rg} -sentence $\exists x : O. \forall f : A(x, x). I(f)$ will be interpreted in an \mathcal{L} -structure \mathcal{M} as the proposition

$$\| \sum_{x : O^{\mathcal{M}}} \prod_{f : A^{\mathcal{M}}(x, x)} I^{\mathcal{M}}(f) \|_1$$

When necessary we will write $\phi^{\mathcal{M}}$ to refer to the interpretation of a certain \mathcal{L} -formula into an \mathcal{L} -structure \mathcal{M} . We can then obtain a notion of *satisfaction* of a formula ϕ in an \mathcal{L} -structure \mathcal{M} and therefore a notion of a *model* \mathcal{M} of an \mathcal{L} -theory \mathbb{T} (where a *theory* \mathbb{T} is a set of \mathcal{L} -sentences). We write **Mod $_{\mathbb{T}}$** for the type of \mathbb{T} -models, i.e. those \mathcal{L} -structures that satisfy all $\phi \in \mathbb{T}$. For example, for the \mathcal{L}_{rg} -theory

$\mathbb{T}_1 = \{\exists x: O.\forall f: A(x, x).I(f)\}$ we have

$$\mathbf{Mod}_{\mathbb{T}_1} =_{\text{df}} \sum_{\mathcal{M}: \mathbf{Struc}_{\mathcal{L}}} \left\| \sum_{x: O^{\mathcal{M}} f: A^{\mathcal{M}}(x, x)} \prod I^{\mathcal{M}}(f) \right\|_1$$

Finally, note that a well-formed context Γ in the sense of FOLDS clearly gives rise to a well-formed context $\Gamma^{\mathcal{M}}$ in type theory, and so we will freely use the notation $\Gamma^{\mathcal{M}}$ to denote the interpretation of Γ in some \mathcal{L} -structure \mathcal{M} .

1.3. Note on Metatheory. Throughout this paper we work in a set-theoretic metatheory. This should already be clear by our use of e.g. the symbol \in in our definition of a FOLDS signature and its associated notions. However, we aim for this set-theoretic metatheory to be “foundation-agnostic” in that our definitions and arguments could equally well be formalized within an extensional set theory like ZF or a suitable model of a structural set theory inside the Univalent Foundations.

One further clarification is in order, to avoid any confusion. Since our metalanguage is set-theoretic we will often refer to expressions in type theory using set-theoretic abbreviations. For example, the symbol

$$\times_{s \in S} R_s$$

might refer to a well-formed type in type theory (a non-dependent sum with S -many components) but is not itself to be understood as a well-formed type in type theory. Rather, it is a description of such a type in our set-theoretic metalanguage (hence the use of “ \in ” under \times). Similarly, in Section 2 below we will construct set-theoretically certain FOLDS formulas which we then use to describe certain well-formed types in type theory. But our FOLDS formulas, as constructed, are not themselves to be understood as constructed *inside* type theory. In particular, our description of the relevant formulas will contain certain distinctions between variables that will be internally invisible to the type theory.

2. A SYNTACTIC DEFINITION OF ISOMORPHISM

Fix a FOLDS signature \mathcal{L} . Let $x: K$ for some $K \in \text{ob}\mathcal{L}$. Then we define the set of *x-compatible* sorts as follows:

$$\mathcal{L}_x =_{\text{df}} \{R \in \text{ob}\mathcal{L} \mid R > K \text{ and } \forall q \in \mathcal{L}(R, K) \forall p_1, p_2 \in \mathcal{L}(K, K'), p_1 q = p_2 q \Rightarrow x_{p_1} = x_{p_2}\}$$

If $R \in \mathcal{L}_x$ we say that x is *R-compatible*. The idea is that a certain variable is not *R-compatible* if its dependent variables violate some identity that R requires in order to be defined. The following example illustrates this.

Example 2.1. In \mathcal{L}_{rg} a variable $f: A$ with $f_d = x$ and $f_c = y$ (and where $x \neq y$) is not *I-compatible*. This is because $di = ci$ yet $f_c \neq f_d$ as just stated. However, in $\mathcal{L}_{\text{rg}}^{\equiv_A}$ f is $=_A$ -compatible (since $=_A$ does not “require” an identification between “source” and “target”).

Remark 2.2. If $\partial x = \partial y$ then $\mathcal{L}_x = \mathcal{L}_y$. In other words, variables with the same boundary are compatible with exactly the same sorts.

The idea of the syntactic definition of isomorphism for K goes as follows. For any variables $x, y: K$ we will define by mutual induction a formula

$$\text{Ind}(x, y) \in \mathbf{Form}_{\mathcal{L}}(\text{dep}(x) \cup \text{dep}(y))$$

that expresses the fact that x and y are “indistinguishable up to equivalence” by any of the sorts they are jointly compatible with (i.e. for all $R \in \mathcal{L}_x \cap \mathcal{L}_y$) and where “equivalence” will be defined, for any $\alpha, \beta: A$, as a formula

$$A(\alpha) \simeq A(\beta) \in \mathbf{Form}_{\mathcal{L}}(\partial\alpha \cup \partial\beta)$$

Our first goal is to define these formulas (by mutual induction).

We begin with the definition of indistinguishability, i.e. of the formula $\text{Ind}(x, y)$. Let $x, y: K$ and $R \in \mathcal{L}_x \cap \mathcal{L}_y$ and $p \in \mathcal{L}(R, K)$. Then we have the following two definitions expressing respectively “ x and y cannot be distinguished by R in position p , up to equivalence” and “ x and y cannot be distinguished by R in any position, up to equivalence”:

$$\begin{aligned} \text{Ind}_R^p(x, y) &=_{\text{df}} \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\alpha) \simeq R(\beta) \mid \alpha, \beta: R, \alpha_p = x, \beta_p = y, \forall p \neq q \in \mathcal{L}(R, K). \alpha_q = \beta_q, \\ &\quad \Gamma_{\alpha, \beta} = \partial\alpha \cup \partial\beta \setminus (\text{dep}(\alpha_p) \cup \text{dep}(\beta_p)) \} / \sim_c \\ \text{Ind}_R(x, y) &=_{\text{df}} \bigwedge_{p \in \mathcal{L}(R, K)} \text{Ind}_R^p(x, y) \end{aligned}$$

Remark 2.3. The use of \sim_c in $\text{Ind}_R^p(x, y)$ ensures that $\text{Ind}_R^p(x, y)$ is finite. Strictly speaking, this makes $\text{Ind}_R^p(x, y)$ a “conjunction” of equivalence classes of formulas up to contextual equivalence. We brush over the difficulty by assuming we in each case choose a canonical representative for the equivalence class.

Example 2.4. Take $\mathcal{L}_{\text{rg}}^=A$. Then we have

$$\text{Ind}_{=A}^s(f, g) = \{ \forall h (f =_A h \simeq g =_A h), f =_A f \simeq g =_A f, f =_A g \simeq g =_A g \}$$

where we have used the standard abbreviation $f =_A g$ for the sort $=_A(f, g)$. Intuitively, $\text{Ind}_{=A}^s(f, g)$ expresses the fact that “binary relation” $=_A$ should not distinguish between f and g when they are “plugged in” position s , i.e. in the “source” position, which we take to be the left-hand side. And “does not distinguish” is in turn understood as the existence of an equivalence (to be defined below) between the sorts $f =_A h$ and $g =_A h$ that we obtain by plugging f and g into the s position, for any h . Clearly this is something that we would expect of the binary relation of equality, although of course at this point this is simply uninterpreted syntax.

Definition 2.5 ($\text{Ind}(x, y), x \cong y$). For $x, y: K$ we define the following formula

$$\text{Ind}(x, y) =_{\text{df}} \bigwedge_{R \in \mathcal{L}_x \cap \mathcal{L}_y} \text{Ind}_R(x, y)$$

which we express as x and y are **indistinguishable**. In the special case when $\partial x = \partial y$ we write $x \cong y$ for $\text{Ind}(x, y)$ and say that x and y are **isomorphic**.

Observe that we have $\text{FV}(\text{Ind}(x, y)) = \text{dep}(x) \cup \text{dep}(y)$. $\text{Ind}(x, y)$ captures the intended meaning that “ x and y are indistinguishable up to equivalence by any x -compatible and y -compatible R , in any position”. Of course, for this definition to be useful we also need to define equivalence as it appears in $\text{Ind}_R^p(x, y)$ (i.e. in the formula $\forall \Gamma_{\alpha, \beta} R(\alpha) \simeq R(\beta)$), which we do below. But even before we do that, we can draw some useful conclusions in some degenerate cases.

Example 2.6. In \mathcal{L}_{rg} let $f, g: A(x, y)$. Then $\mathcal{L}_f = \mathcal{L}_g = \emptyset$ which means $f \cong g \equiv_{\alpha} \top$ (as the empty conjunction). This is a strange but I think correct conclusion. For if two terms cannot be distinguished by any data in the FOLDS signature (i.e. they have no compatible sorts) then the appropriate saturation condition is to regard them as always isomorphic.

Example 2.7. A bit stranger is the situation where, say, $f: A(x, x)$ and $g: A(x, y)$, still in \mathcal{L}_{rg} . In this case, we have $\mathcal{L}_f \neq \emptyset$, but $\mathcal{L}_f \cap \mathcal{L}_g = \emptyset$, and so Definition 2.5 dictates that we should regard f and g as always indistinguishable. Formally, this does not seem to be problematic but it is a little harder to make sense of since there is certainly something in the syntax that distinguishes f and g , namely that in the case of the former we have $f_d = f_c$ but the analogous equation does not hold of g .

Example 2.8. In $\mathcal{L}_{\text{rg}}^{\equiv_A}$, let $f, g: A(x, y)$. Then $\mathcal{L}_f = \mathcal{L}_g = \{=_A\}$ and we have

$$\begin{aligned} \text{Ind}(f, g) &= \text{Ind}_{=_A}^s(f, g) \wedge \text{Ind}_{=_A}^t(f, g) \\ &= \{\forall h(f =_A h \simeq g =_A h), f =_A f \simeq g =_A f, f =_A g \simeq g =_A g\} \wedge \\ &\quad \{\forall h(h =_A f \simeq h =_A g), f =_A f \simeq f =_A g, g =_A f \simeq g =_A g\} \end{aligned}$$

Proposition 2.9. For $K \in \text{ob}\mathcal{L}$, $l(K) = 1$, $x, y: K$ we have $\text{Ind}(x, y) \equiv_{\alpha} \top$.

Proof. We have $\mathcal{L}_x = \mathcal{L}_y = \emptyset$ hence $\text{Ind}(x, y)$ is an empty conjunction and therefore the same thing as \top . \square

We now turn our attention to the definition of \simeq .

Definition 2.10 ($K(\alpha) \simeq K(\beta)$). For $K \in \text{ob}\mathcal{L}$ and $\alpha, \beta: K$ we define the following formula:

$$\begin{aligned} K(\alpha) \simeq K(\beta) =_{\text{df}} & \left[\forall x: K(\alpha) \exists y: K(\beta). \text{Ind}(x, y) \wedge (\forall y': K(\beta). \text{Ind}(x, y') \rightarrow y \cong y') \right] \wedge \\ & \left[\forall x, x': K(\alpha) \forall y, y': K(\beta). \text{Ind}(x, y) \wedge \text{Ind}(x', y') \wedge y \cong y' \rightarrow x \cong x' \right] \wedge \\ & \left[\forall y: K(\beta) \exists x: K(\alpha). \text{Ind}(x, y) \right] \end{aligned}$$

When the formula holds we say that $K(\alpha)$ is **equivalent** to $K(\beta)$.

To aid understanding, here is the intended meaning behind $K(\alpha) \simeq K(\beta)$: the first conjunct expresses that $\text{Ind}(x, y)$ is a functional relation (up to \cong), the second line expresses that $\text{Ind}(x, y)$ is injective (up to \cong), and the third line express that $\text{Ind}(x, y)$ is surjective. Putting all this together, $K(\alpha) \simeq K(\beta)$ can be understood as saying “ $\text{Ind}(x, y)$ is an equivalence up to \cong ”.

Note that we have $\text{FV}(K(\alpha) \simeq K(\beta)) = \partial\alpha \cap \partial\beta$. Indeed, since $K(\alpha) \simeq K(\beta)$ depends only on the respective boundaries of α and β it is usually helpful to write $K(\partial\alpha) \simeq K(\partial\beta)$ when the variables in the boundary can be explicitly listed. For example, we can write $A(x, y)$ for the sort of “arrows” in \mathcal{L}_{rg} dependent on $x, y: O$ instead of the more alienating $A(f)$ (for some f that depends on x and y).

Example 2.11. In \mathcal{L}_{rg} let $x, y: O$. Then we have:

$$\begin{aligned}
A(x, x) \simeq A(y, y) = & \\
& \left[\forall f: A(x, x) \exists g: A(y, y). \text{Ind}(f, g) \wedge (\forall h: A(y, y). \text{Ind}(f, h) \rightarrow g \cong h) \right] \wedge \\
& \left[\forall f, f': A(x, x) \forall g, g': A(y, y). \text{Ind}(f, g) \wedge \text{Ind}(f', g') \wedge g \cong g' \rightarrow f \cong f' \right] \wedge \\
& \left[\forall g: A(y, y) \exists f: A(x, x). \text{Ind}(f, g) \right] = \\
& \left[\forall f: A(x, x) \exists g: A(y, y). (I(f) \simeq I(g)) \wedge (\forall h: A(y, y). (I(f) \simeq I(h)) \right. \\
& \quad \left. \rightarrow (I(g) \simeq I(h))) \right] \wedge \\
& \left[\forall f, f': A(x, x) \forall g, g': A(y, y). (I(f) \simeq I(g)) \wedge (I(f') \simeq I(g')) \wedge (I(g) \simeq I(g')) \right. \\
& \quad \left. \rightarrow (I(f) \simeq I(f')) \right] \wedge \\
& \left[\forall g: A(y, y) \exists f: A(x, x). (I(f) \simeq I(g)) \right]
\end{aligned}$$

The only sort that could distinguish two variables $f: A(x, x)$ and $g: A(y, y)$ in \mathcal{L}_{rg} is I . Therefore, $\text{Ind}(f, g)$ becomes $(I(f) \simeq I(g))$ as prescribed by the definition of Ind . So what we get, intuitively, is that $A(x, x)$ and $A(y, y)$ are equivalent iff there is a one-to-one correspondence between “terms” f of $A(x, x)$ and “terms” g of $A(y, y)$ that give equivalent sorts when plugged into I , i.e. such that $I(f) \simeq I(g)$. Except this “one-to-one correspondence” is one-to-one only up to \cong which in the case of \mathcal{L}_{rg} amounts to the exact same property, namely $g \cong h$ iff g and h produce equivalent sorts when plugged into the “type family” I . We clearly here want to regard I as a predicate (e.g. picking out which “arrows” are “identities”) which would make the equivalence $I(f) \simeq I(g)$ a logical equivalence. This is what the next proposition establishes for sorts, like I , of level 1.

Proposition 2.12. For $K \in \text{ob}\mathcal{L}$, $l(K) = 1$ we have $K(\alpha) \simeq K(\beta) \Leftrightarrow K(\alpha) \leftrightarrow K(\beta)$

Proof. Note that $\mathcal{L}_\alpha = \mathcal{L}_\beta = \emptyset$ which means that for $x, x': K(\alpha)$ and $y', y: K(\beta)$ we have

$$\text{Ind}_R^p(x, y) \Leftrightarrow \text{Ind}_R^p(x', y) \Leftrightarrow \text{Ind}_R^p(x, y') \Leftrightarrow x \cong x' \Leftrightarrow y \cong y' \Leftrightarrow \top$$

Therefore, by Proposition 2.9, we can see that $K(\alpha) \simeq K(\beta)$ is logically equivalent to

$$\forall x: K(\alpha) \exists y: K(\beta). \top \wedge \forall y: K(\beta) \exists x: K(\alpha). \top$$

which in turn is logically equivalent to $K(\alpha) \leftrightarrow K(\beta)$. \square

Using propositions 2.9 and 2.12 we can successively obtain formulas for $x \cong y$ for $x, y: K$ for sorts K or arbitrarily high level. We will write \cong_K for the isomorphism relation defined on a sort K , or $\cong_{K(\alpha)}$ whenever we want to make the specific context explicit. We now prove the following crucial result, which is the last general result we will carry out “formally”, i.e. assuming an arbitrary formal system for FOLDS.

Proposition 2.13. *Let \mathcal{L} be a FOLDS signature. For any $K \in \text{ob}\mathcal{L}$, $\text{Ind}(x, y)$ is an equivalence relation. That is, it satisfies the following three properties:*

$$\begin{aligned} (\text{refl}_{\text{Ind}}) \quad & \forall x: K. \text{Ind}(x, x) \\ (\text{sym}_{\text{Ind}}) \quad & \forall x, y: K. \text{Ind}(x, y) \Rightarrow \text{Ind}(y, x) \\ (\text{tran}_{\text{Ind}}) \quad & \forall x, y, z: K. \text{Ind}(x, y) \wedge \text{Ind}(y, z) \Rightarrow \text{Ind}(x, z) \end{aligned}$$

In particular, for any $K \in \text{ob}\mathcal{L}$, $x \cong y$ is an equivalence relation.

Proof. We proceed by induction on the level of K .

Base Step: For any sort K with $l(K) = 1$ the statement follows immediately from Proposition 2.9.

Inductive Step: For any choices of α, β, γ define the following formulas:

$$\begin{aligned} (\text{refl}_{\simeq}) \quad & K(\alpha) \simeq K(\alpha) \\ (\text{sym}_{\text{Ind}}) \quad & K(\alpha) \simeq K(\beta) \Rightarrow K(\beta) \simeq K(\alpha) \\ (\text{tran}_{\text{Ind}}) \quad & K(\alpha) \simeq K(\beta) \wedge K(\beta) \simeq K(\gamma) \Rightarrow K(\alpha) \simeq K(\gamma) \end{aligned}$$

Let us denote the conjunction of the three statements above as (eqr_{\simeq}^K) and similarly let us denote by $(\text{eqr}_{\text{Ind}}^K)$ the conjunction of the three sentences in the statement of the Proposition. Similarly, let us write $(\text{eqr}_{\text{Ind}}^n)$ for the statement that for any sort K with $l(K) \geq n$, $(\text{eqr}_{\text{Ind}}^K)$ holds and (eqr_{\simeq}^n) for the analogous statement for \simeq . The proof of the inductive step now proceeds in two steps. First we prove that $(\text{eqr}_{\text{Ind}}^n) \Rightarrow (\text{eqr}_{\simeq}^n)$ and then that $(\text{eqr}_{\simeq}^n) \Rightarrow (\text{eqr}_{\text{Ind}}^{n+1})$. Put together these give $(\text{eqr}_{\text{Ind}}^n) \Rightarrow (\text{eqr}_{\text{Ind}}^{n+1})$ which is exactly what the inductive step requires us to prove. We take them in turn.

$(\text{eqr}_{\text{Ind}}^n) \Rightarrow (\text{eqr}_{\simeq}^n)$: For (refl_{\simeq}) we have:

$$\begin{aligned}
 (\text{refl}_{\simeq}) &=_{\text{df}} K(\alpha) \simeq K(\alpha) \\
 &=_{\text{df}} \left[\forall x: K(\alpha) \exists y: K(\alpha). \text{Ind}(x, y) \wedge (\forall y': K(\alpha). \text{Ind}(x, y') \rightarrow y \cong y') \right] \wedge \\
 &\quad \left[\forall x, x': K(\alpha) \forall y, y': K(\alpha). \text{Ind}(x, y) \wedge \text{Ind}(x', y') \wedge y \cong y' \rightarrow x \cong x' \right] \wedge \\
 &\quad \left[\forall y: K(\alpha) \exists x: K(\alpha). \text{Ind}(x, y) \right] \\
 &\equiv_{\alpha} \left[\forall x: K(\alpha) \exists y: K(\alpha). x \cong y \wedge (\forall y': K(\alpha). x \cong y' \rightarrow y \cong y') \right] \wedge \\
 &\quad \left[\forall x, x': K(\alpha) \forall y, y': K(\alpha). x \cong y \wedge x' \cong y' \wedge y \cong y' \rightarrow x \cong x' \right] \wedge \\
 &\quad \left[\forall y: K(\alpha) \exists x: K(\alpha). x \cong y \right]
 \end{aligned}$$

The last form is obtained because when x, y are of the same sort we write $x \cong y$ instead of $\text{Ind}(x, y)$. The first conjunct $\forall x: K(\alpha) \exists y: K(\alpha). x \cong y \wedge (\forall y': K(\alpha). x \cong y' \rightarrow y \cong y')$ follows from the reflexivity of \cong , which is given by the inductive hypothesis. The second conjunct $\forall x, x': K(\alpha) \forall y, y': K(\alpha). x \cong y \wedge x' \cong y' \wedge y \cong y' \rightarrow x \cong x'$ follows from the symmetry and transitivity of \cong , which is once again given by the inductive hypothesis. Finally, the third conjunct $\forall y: K(\alpha) \exists x: K(\alpha). x \cong y$ follows from the reflexivity of \cong which is once again given by the inductive hypothesis. For (sym_{\simeq}) we have:

$$\begin{aligned}
 (\text{sym}_{\simeq}) &=_{\text{df}} K(\alpha) \simeq K(\beta) \Rightarrow K(\beta) \simeq K(\alpha) \\
 &=_{\text{df}} \left[\forall x: K(\alpha) \exists y: K(\beta). (\text{Ind}(x, y) \wedge (\forall y': K(\beta). \text{Ind}(x, y') \rightarrow y \cong y')) \wedge \right. \\
 &\quad \left. \forall x, x': K(\alpha) \forall y, y': K(\beta). \text{Ind}(x, y) \wedge \text{Ind}(x', y') \wedge y \cong y' \rightarrow x \cong x' \wedge \right. \\
 &\quad \left. \forall y: K(\beta) \exists x: K(\alpha). \text{Ind}(x, y) \right] \Rightarrow \\
 &\quad \left[\forall x: K(\beta) \exists y: K(\alpha). \text{Ind}(x, y) \wedge (\forall y': K(\alpha). \text{Ind}(x, y') \rightarrow y \cong y') \wedge \right. \\
 &\quad \left. \forall x, x': K(\beta) \forall y, y': K(\alpha). \text{Ind}(x, y) \wedge \text{Ind}(x', y') \wedge y \cong y' \rightarrow x \cong x' \wedge \right. \\
 &\quad \left. \forall y: K(\alpha) \exists x: K(\beta). \text{Ind}(x, y) \right]
 \end{aligned}$$

By the inductive hypothesis, we can assume that Ind is symmetric, which means that if $\text{Ind}(x, y)$ defines an “equivalence up to \cong ” then $\text{Ind}(y, x)$ also defines such an equivalence. We can use this basic idea to deduce each of the conjuncts of the RHS from the LHS. We deduce the conjunct $\forall y: K(\alpha) \exists x: K(\beta). \text{Ind}(x, y)$ on the RHS as

an illustration. We have:

$$\begin{aligned}
 & K(\alpha) \simeq K(\beta) \Rightarrow \forall y: K(\beta) \exists x: K(\alpha). \text{Ind}(x, y) \\
 (\text{sym}) \quad & \Rightarrow \forall y: K(\beta) \exists x: K(\alpha). \text{Ind}(y, x) \\
 & \equiv_{\alpha} \forall x: K(\beta) \exists y: K(\alpha). \text{Ind}(x, y)
 \end{aligned}$$

By an exactly analogous argument we can establish (tran_{\simeq}) .

$(\text{eqr}_{\text{Ind}}^n \Rightarrow \text{eqr}_{\text{Ind}}^{n+1})$: Let K be a sort with $l(K) = n+1$. Firstly, note that it suffices to establish $(\text{eqr}_{\text{Ind}_R^p})$ for any $R > K$ and $p: R \rightarrow K$ since Ind is a conjunction over all such R and p . Thus, the properties to be shown are the following:

$$\begin{aligned}
 (\text{refl}_{\text{Ind}_R^p}) \quad & \forall x: K. \text{Ind}_R^p(x, x) \\
 (\text{sym}_{\text{Ind}_R^p}) \quad & \forall x, y: K. \text{Ind}_R^p(x, y) \Rightarrow \text{Ind}_R^p(y, x) \\
 (\text{tran}_{\text{Ind}_R^p}) \quad & \forall x, y, z: K. \text{Ind}_R^p(x, y) \wedge \text{Ind}_R^p(y, z) \Rightarrow \text{Ind}_R^p(x, z)
 \end{aligned}$$

We take each of the three statements to be shown in turn. Firstly, for $(\text{refl}_{\text{Ind}_R^p})$ we have:

$$\begin{aligned}
 \text{Ind}_R^p(x, x) &=_{\text{df}} \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\alpha) \simeq R(\beta) \mid \alpha, \beta: R, \alpha_p = x, \beta_p = x, \forall q \neq p (\alpha_q = \beta_q) \} / \sim_c \\
 &= \bigwedge \{ \forall \Gamma_{\alpha, \alpha} R(\alpha) \simeq R(\alpha) \mid \alpha: R, \alpha_p = x \} / \sim_c
 \end{aligned}$$

where the equality follows since α and β in the general definition of $\text{Ind}_R^p(x, y)$ are stipulated to differ only in the variable they assign to position p , which is here going to be the same variable. We therefore get that $\text{Ind}_R^p(x, x)$ holds iff $R(\alpha) \simeq R(\alpha)$ holds, and we know that the latter is the case from the inductive hypothesis together with the first part of the inductive step above, since $R > K$.

Secondly, for $(\text{sym}_{\text{Ind}_R^p})$ we have:

$$\begin{aligned}
 \text{Ind}_R^p(x, y) &=_{\text{df}} \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\alpha) \simeq R(\beta) \mid \alpha, \beta: R, \alpha_p = x, \beta_p = y, \forall p \neq q (\alpha_q = \beta_q) \} / \sim_c \\
 (\text{sym}_{\simeq}) \quad & \Rightarrow \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\beta) \simeq R(\alpha) \mid \alpha, \beta: R, \alpha_p = x, \beta_p = y, \forall p \neq q (\alpha_q = \beta_q) \} / \sim_c \\
 & \equiv_{\alpha} \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\beta) \simeq R(\alpha) \mid \beta, \alpha: R, \beta_p = x, \alpha_p = y, \forall p \neq q (\alpha_q = \beta_q) \} / \sim_c \\
 &=_{\text{df}} \text{Ind}_R^p(y, x)
 \end{aligned}$$

Thirdly, for $(\text{tran}_{\text{Ind}_R^p})$ we have:

$$\begin{aligned}
& \text{Ind}_R^p(x, y) \wedge \text{Ind}_R^p(y, z) \\
&=_{\text{df}} \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\alpha) \simeq R(\beta) \mid \alpha, \beta: R, \alpha_p = x, \beta_p = y, \forall p \neq q (\alpha_q = \beta_q) \} / \sim_c \\
&\quad \wedge \bigwedge \{ \forall \Gamma_{\alpha, \beta} R(\alpha) \simeq R(\beta) \mid \alpha, \beta: R, \alpha_p = y, \beta_p = z, \forall p \neq q (\alpha_q = \beta_q) \} / \sim_c \\
&\Leftrightarrow \bigwedge \{ \forall \Gamma_{\alpha, \beta} \cup \Gamma_{\beta, \gamma} R(\alpha) \simeq R(\beta) \wedge R(\beta) \simeq R(\gamma) \mid \alpha_p = x, \beta_p = y, \gamma_p = z \} / \sim_c \\
&\Rightarrow \bigwedge \{ \forall \Gamma_{\alpha, \gamma} R(\alpha) \simeq R(\gamma) \mid \alpha_p = x, \gamma_p = z \} / \sim_c \\
&=_{\text{df}} \text{Ind}_R^p(x, z)
\end{aligned}$$

This establishes the fact that $\text{Ind}(x, y)$ is an equivalence relation, thus completing the inductive step. \square

We conclude this section by comparing our syntactic definitions to the usual notions. In \mathcal{L}_{cat} , for $x, y: O$ let $\text{Iso}(x, y)$ denote the formula expressing that x and y are isomorphic in the traditional sense (“There exist mutually inverse $f: A(x, y)$ and $g: A(y, x)$ ”) Call this notion *categorical isomorphism*. The following proposition says that in \mathbb{T}_{cat} the two notions of isomorphism are logically equivalent.

Proposition 2.14. $\mathbb{T}_{\text{cat}} \models \text{Iso}(x, y) \Leftrightarrow x \cong y$

Proof. Firstly, note that by the axioms for equality on A we will get that for any $f, f': A(x, y)$

$$\mathbb{T}_{\text{cat}} \models f \cong_A f' \Leftrightarrow f =_A f'$$

This allows us to replace (up to logical equivalence) the $=_A$ as it appears in $\text{Iso}(x, y)$ with our notion of \cong_A . We will assume we have done so. The (\Rightarrow) direction follows from the well-known result (proven e.g. in [Mak95]) that two categorically isomorphic objects in a category satisfy exactly the same \mathcal{L}_{cat} -formulas, and are therefore indistinguishable in exactly the sense of $x \cong y$. For the (\Leftarrow) direction, note that from the Yoneda Lemma we know that x and y are categorically isomorphic if and only if their images under the Yoneda embedding are isomorphic, i.e. if we have a natural isomorphism from $\mathbf{y}(-, x)$ to $\mathbf{y}(-, y)$. The existence of such a natural isomorphism can be expressed as the following \mathcal{L}_{cat} -formula

$$\begin{aligned}
\text{Yso}(x, y) =_{\text{df}} & \forall z, w: O \forall f: A(w, z) \forall g: A(z, x) \forall h: A(w, x) \forall k: A(w, y) \forall l: A(z, y) \\
& A(z, x) \simeq A(z, y) \wedge (\circ(g, f, h) \wedge \text{Ind}(h, k) \leftrightarrow \text{Ind}(g, l) \wedge \circ(l, f, k))
\end{aligned}$$

By the Yoneda Lemma we have $\mathbb{T}_{\text{cat}} \models \forall x \forall y \text{Iso}(x, y) \Leftrightarrow \text{Yso}(x, y)$. But $\text{Yso}(x, y)$ is clearly a consequence of $x \cong y$ since both conjuncts appear in the definitions given above if, using Proposition 2.12, we regard \leftrightarrow as an instance of \simeq . We thus get $x \cong y \Rightarrow \text{Iso}(x, y)$ \square

Remark 2.15. An important difference with our general notion \cong and the usual notions of isomorphism is that $x \cong y$ is not necessarily “witnessed” by some other variable, e.g. an arrow f as in $\text{Iso}(x, y)$. Thus, at this level, it is not possible to define a *set* of isomorphisms $x \cong y$, but rather only a *proposition* expressing whether or not x and y are isomorphic. To overcome this limitation, in Section 3 we will interpret

the existential quantifiers as untruncated Σ -types, which will recover in the semantics the intuitively correct h -level.

3. SATURATION IN TYPE THEORY

We now interpret our syntactic definition of isomorphism in type theory. However, in order to interpret $x \cong y$ as defined in Section 2 we make one crucial change to the usual interpretation, which is that we will not truncate the existential quantifiers that appear in the definition of \simeq in Definition 2.10 and interpret them instead as full Σ -types. This is made precise in the following definition.

Definition 3.1 (Interpretation of \cong in type theory). Let \mathcal{L} be a FOLDS signature and \mathcal{M} an \mathcal{L} -structure. For $K \in \text{ob}\mathcal{L}$ and $\alpha, \beta: K^{\mathcal{M}}$ we define the following type (in context $(\partial\alpha \cup \partial\beta)^{\mathcal{M}}$):

$$K(\alpha) \simeq^{\mathcal{M}} K(\beta) =_{\text{df}} \left[\prod_{x: K^{\mathcal{M}}(\alpha)} \sum_{y: K^{\mathcal{M}}(\beta)} \text{Ind}^{\mathcal{M}}(x, y) \times \left(\prod_{y': K^{\mathcal{M}}(\beta)} \text{Ind}^{\mathcal{M}}(x, y') \rightarrow y \cong^{\mathcal{M}} y' \right) \right] \times \\ \left[\prod_{\substack{x, x': K^{\mathcal{M}}(\alpha) \\ y, y': K^{\mathcal{M}}(\beta)}} \left(\text{Ind}^{\mathcal{M}}(x, y) \times \text{Ind}^{\mathcal{M}}(x', y') \times y \cong^{\mathcal{M}} y' \rightarrow x \cong^{\mathcal{M}} x' \right) \right] \times \\ \left[\prod_{y: K^{\mathcal{M}}(\beta)} \sum_{x: K^{\mathcal{M}}(\alpha)} \text{Ind}^{\mathcal{M}}(x, y) \right]$$

By mutual induction we also define for $x, y: K^{\mathcal{M}}$ the following type (in context $(\text{dep}(x) \cup \text{dep}(y))^{\mathcal{M}}$):

$$\text{Ind}^{\mathcal{M}}(x, y) =_{\text{df}} \times_{\substack{R \in \mathcal{L}_x \cap \mathcal{L}_y \\ p \in \mathcal{L}(R, K)}} \left\{ \prod_{\mathbf{z}: \Gamma_{\alpha, \beta}^{\mathcal{M}}} R(\alpha) \simeq^{\mathcal{M}} R(\beta) \mid \alpha_p = x, \beta_p = y, \forall q \neq p (\alpha_q = \beta_q) \right\} / \sim_c$$

Whenever $x, y: K^{\mathcal{M}}$ are such that $\partial x = \partial y$ we write $x \cong^{\mathcal{M}} y$ for $\text{Ind}^{\mathcal{M}}(x, y)$. Thus, given any FOLDS signature \mathcal{L} , any $K \in \text{ob}\mathcal{L}$ we obtain a relation (in an appropriate context):

$$\cong_{K^{\mathcal{M}}}: K^{\mathcal{M}} \times K^{\mathcal{M}} \rightarrow \mathcal{U}$$

Notation. Whenever the sort of x and y is understood we will usually denote the type $\cong_{K^{\mathcal{M}}}(x, y)$ as simply $x \cong y$.

Let \mathcal{M} be an \mathcal{L} -structure and K any sort in \mathcal{L} , with $K^{\mathcal{M}}$ its interpretation. Proposition 2.13 holds also in type theory (since the proof was purely formal) and so the relation $\cong_{K^{\mathcal{M}}}$ will be reflexive. In particular, this means that given any context Γ appropriate to K we have

$$\Gamma^{\mathcal{M}}, x: K^{\mathcal{M}} \vdash \rho_x: x \cong x$$

where ρ_x is some term witnessing the reflexivity. This gives us a canonical map

$$\Gamma^{\mathcal{M}}, x, y: K^{\mathcal{M}} \vdash \text{idtoiso}_{x, y}: x = y \rightarrow x \cong y$$

by the usual induction on identity.

Definition 3.2 (Saturation). Let \mathcal{L} be a FOLDS signature, $K \in \text{ob}\mathcal{L}$, Γ a context appropriate to K and \mathcal{M} an \mathcal{L} -structure. Then the **saturation for $K^{\mathcal{M}}$** is the following proposition (in context $\Gamma^{\mathcal{M}}$):

$$\mathbf{Sat}(K^{\mathcal{M}}) =_{\text{df}} \prod_{x,y: K^{\mathcal{M}}} \text{isequiv}(\text{idtoiso}_{x,y})$$

For an \mathcal{L} -structure \mathcal{M} , we say that K is **saturated** (or that \mathcal{M} is **K -saturated**) if $\mathbf{Sat}(K^{\mathcal{M}})$ is inhabited. We will say that an \mathcal{L} -structure is **saturated at level n** (or **n -saturated**) if it is K -saturated for any $K \in \text{ob}\mathcal{L}$ with $l(K) \leq n$. We will also say that an \mathcal{L} -structure \mathcal{M} is **totally saturated** if it is saturated at every level. We write $\mathbf{SatStr}_{\mathcal{L}}$ (resp. $\mathbf{SatStr}_{\mathcal{L}}^n$) for the type of totally saturated (resp. n -saturated) \mathcal{L} -structures and $\mathbf{SatMod}_{\mathbb{T}}$ (resp. $\mathbf{SatMod}_{\mathbb{T}}^n$) for the type of totally saturated (resp. n -saturated) \mathbb{T} -models (for some \mathcal{L} -theory \mathbb{T}).

Based on Definition 3.2 we regard $\mathbf{Struc}_{\mathcal{L}}$ (resp. $\mathbf{Mod}_{\mathbb{T}}$) as the type of “totally unsaturated” types of \mathcal{L} -structures (resp. models of \mathbb{T}). The following example illustrates the idea.

Example 3.3.

$$\begin{aligned} \mathbf{Struc}_{\mathcal{L}_{\text{rg}}} &=_{\text{df}} \sum_{\substack{O: \mathcal{U} \\ A: O \times O \rightarrow \mathcal{U}}} \left(\prod_{x: A} A(x, x) \rightarrow \mathcal{U} \right) \\ \mathbf{SatStruc}_{\mathcal{L}_{\text{rg}}}^1 &\simeq \sum_{\substack{O: \mathcal{U} \\ A: O \times O \rightarrow \mathcal{U}}} \prod_{x: A} (A(x, x) \rightarrow \mathbf{Prop}_{\mathcal{U}}) \end{aligned}$$

It is important to note the difference between $=_{\text{df}}$ and \simeq above. The RHS in the first line is the definition of $\mathbf{Struc}_{\mathcal{L}_{\text{rg}}}$ whereas the RHS of the second line is a consequence of Proposition 2.9.

The most relevant notion for practical purposes is saturation at a particular level (or total saturation). This is because K -saturation for an individual K may be of very little use unless we know that everything that depends on K is already saturated. For example, it is not of much use to state the saturation condition for objects in a precategory unless we already know that the hom-sets are h -sets. On the contrary, if we know that \mathcal{L} is saturated at level n and K is a sort of level n , then K -saturation means that isomorphism in K is expressed in terms of the identity types of sorts of lower level, since these have already been declared equivalent to isomorphism. This has the consequence that saturation at a particular level correlates very well with h -levels as the following proposition establishes.

Proposition 3.4. *Let \mathcal{L} be a FOLDS signature and \mathcal{M} an \mathcal{L} -structure saturated at level n . Then for any $K \in \text{ob}\mathcal{L}$ with $l(K) = n$, $K^{\mathcal{M}}$ is of h -level n .*

Proof. We proceed by induction on the level of K . For the case $l(K) = 1$ note that \mathcal{M} is 1-saturated if and only if for any sort K of level 1, the map

$$\text{idtoiso}_{x,y}: x =_{K^{\mathcal{M}}} y \rightarrow (x \cong y)^{\mathcal{M}}$$

is an equivalence. But by Proposition 2.9 we have $(x \cong^{\mathcal{M}} y) \equiv_{\alpha} \top$ and hence $\text{idtoiso}_{x,y}$ is the unique map into $\mathbf{1}$. Therefore, we have

$$\text{Sat}(K^{\mathcal{M}}) \simeq \prod_{x,y: K^{\mathcal{M}}} x =_{K^{\mathcal{M}}} y$$

and the RHS is equivalent to $\text{isprop}(K^{\mathcal{M}})$. Therefore, K is saturated in \mathcal{M} if and only if $K^{\mathcal{M}}$ is of h -level 1. Now assume that the proposition holds for n and let K be a sort of level $n+1$ and \mathcal{M} an \mathcal{L} -structure saturated at level $n+1$. Saturation for K asserts the equivalence between the identity type on K with $x \cong_K y$. But $x \cong y$, as a type dependent on $K^{\mathcal{M}}$, consists of the following (non-dependent) product:

$$\prod_{R \in \mathcal{L}_x \cap \mathcal{L}_y} \prod_{p: R \rightarrow K} (\text{Ind}_R^p(x, y))^{\mathcal{M}}$$

But Ind_R^p is itself a non-dependent product of Π -types of the form

$$\prod_{\alpha, \beta} [R(\alpha) \simeq^{\mathcal{M}} R(\beta)]$$

Since $R^{\mathcal{M}}(\beta)$ and $R^{\mathcal{M}}(\alpha)$ are of h -level n by the inductive hypothesis, then by the explicit form of $R(\alpha) \simeq^{\mathcal{M}} R(\beta)$ (cf. Definition 3.1) we also get that $R(\alpha) \simeq^{\mathcal{M}} R(\beta)$ is of h -level n . Since (dependent or non-dependent) products preserve h -level we get that $(x \cong y)^{\mathcal{M}}$ is of h -level n . Thus, if $K^{\mathcal{M}}$ is saturated, its identity types will be of h -level n , and thus $K^{\mathcal{M}}$ will itself be of h -level $n+1$, as required. \square

The following lemma is also an easy consequence of our definitions. It makes precise in what way the syntactically defined notion of equivalence becomes a “functional” notion in the semantics, as long as everything “above” has been saturated.

Lemma 3.5. *Let \mathcal{L} be a FOLDS signature and \mathcal{M} an \mathcal{L} -structure saturated at level n . Then for any $K \in \text{ob}\mathcal{L}$ with $l(K) = n$ we have:*

$$[K(\alpha) \simeq^{\mathcal{M}} K(\beta)] \simeq \sum_{f: K^{\mathcal{M}}(\alpha) \rightarrow K^{\mathcal{M}}(\beta)} \text{bij}(f) \times \prod_{x: K^{\mathcal{M}}(\alpha)} \text{Ind}^{\mathcal{M}}(x, f(x))$$

Proof. By inspecting the definition of we can see that the first type in $K(\alpha) \simeq^{\mathcal{M}} K(\beta)$ is equivalent to the type of functions $K^{\mathcal{M}}(\alpha) \rightarrow K^{\mathcal{M}}(\beta)$, since $\cong^{\mathcal{M}}$ can simply be replaced by $=_{K^{\mathcal{M}}}$ since we are assuming \mathcal{M} is n -saturated. And the last two types state exactly that this function is injective and (split) surjective, and hence bijective. \square

Remark 3.6. Saturation as defined by Definition 3.2 relies essentially on the type-theoretic semantics of FOLDS. This means that saturated models of \mathbb{T} are defined “semantically” since we have not provided a way of expressing the condition of saturation in the syntax of FOLDS. One might therefore wonder if there is a way to define saturation in an entirely syntactic manner in FOLDS, and then obtain the usual saturated notions by interpreting that definition. This can indeed be done as long as we syntactically define “transport bijections” that for each “inhabitant” of $x \cong_K y$ yield an “equivalence” between $A(x, \mathbf{w})$ and $A(y, \mathbf{w})$ for any A dependent on K . We do this in full generality in [Tse] for the purposes of a completeness proof, but this machinery is somewhat independent of the purposes of this paper.

4. A HIGHER STRUCTURE IDENTITY PRINCIPLE

A Structure Identity Principle in the sense of [Uni13] asserts something along the lines of: “Isomorphism between structures of a certain kind is equivalent to identity.” In [Uni13] such a statement was proven for set-level structures, i.e. for first-order structures in the traditional sense. Here we prove a generalized Structure Identity Principle where by “structure” we will understand saturated \mathcal{L} -structures for (finite) FOLDS signatures \mathcal{L} of height 3 and by “isomorphic” we will understand the pre-existing notion of FOLDS equivalence, suitably internalized in type theory.

Let us first define FOLDS equivalence. Fix an \mathcal{L} and let us for the time being work “extrenally” i.e. in a traditional set-theoretic set-up, writing M, N, \dots instead of $\mathcal{M}, \mathcal{N}, \dots$ to distinguish the two notions of \mathcal{L} -structure. An \mathcal{L} -structure is a functor $M: \mathcal{L} \rightarrow \mathbf{Set}$ and a *homomorphism* of \mathcal{L} -structures is a natural transformation between the corresponding functors. Write y for the Yoneda embedding and \hat{y} for subfunctor of y that misses the identity on the given object. For $K \in \text{ob}\mathcal{L}$ a K -boundary of an \mathcal{L} -structure M is a natural transformation $\delta: \hat{y}K \Rightarrow M$ and we define the *fiber* of $M(K)$ over δ as the set $M(K)[\delta] = \{a \in M(K): \partial a = \delta\}$. Now let M, N be \mathcal{L} -structures and let $f: M \rightarrow N$ be an \mathcal{L} -structure homomorphism and let δ be a K -boundary of M . This induces a K -boundary $f \circ \delta$ of N by composition with f and thereby induces a map f_δ on the fibers that takes $a \mapsto f_K(a)$. (The fact that this is well-defined follows from the naturality of f .) We now say that f is *fiberwise surjective* if and only if for all K in \mathcal{L} and all K -boundaries δ the map $m_\delta: M(K)[\delta] \rightarrow M(K)[m \circ \delta]$ is surjective.

Definition 4.1 ([Mak95]). We say that M and N are *FOLDS \mathcal{L} -equivalent* if there exists an \mathcal{L} -structure P and fiberwise surjective homomorphisms $m: P \rightarrow M$ and $n: P \rightarrow N$.

We now internalize this definition in type theory. To even write it down, however, we will now restrict ourselves to signatures \mathcal{L} of height 3 and we will assume that $\mathbf{Str}_{\mathcal{L}}$ is given by the Σ -type where the relations in \mathcal{L} have been “written in”. To be more precise, we will adopt the following notational conventions. For the rest of the paper \mathcal{L} is a FOLDS signature of height 3.

Notation. We denote the objects in \mathcal{L} of level 3 by $\mathbf{O} = O_1, \dots, O_n$, those of level 2 by $\mathbf{A} = A_1, \dots, A_m$ and those of level 1 by $\mathbf{R} = R_1, \dots, R_k$. We will abbreviate a sequence of variable declarations $x_1: O_1, \dots, x_n: O_n$ as $\mathbf{x}: \mathbf{O}$, leaving the indexing implicit. Similarly, we will abbreviate a product $A_1 \rightarrow B_1 \times \dots \times A_n \rightarrow B_n$ as $\mathbf{A} \rightarrow \mathbf{B}$. In general, whenever there is any boldface character in our definitions below, this will mean that the definition is taken over the whole (non-dependent) product that the implicit indexing suggests. For example, an expression of the form $\prod_{\mathbf{x}: \mathbf{O}} T(m_{\mathbf{O}}(\mathbf{x}))$ is an abbreviation for

$$\prod_{x_1: O_1, \dots, x_n: O_n} \left[T(m_{O_1}(x_1, \dots, x_n)) \times \dots \times T(m_{O_n}(x_1, \dots, x_n)) \right]$$

Definition 4.2 (\mathcal{L} -homomorphism). Let \mathcal{M} and \mathcal{N} be \mathcal{L} -structures. The type of **\mathcal{L} -homomorphisms** between \mathcal{M} and \mathcal{N} is defined as

$$\text{Hom}_{\mathcal{L}}(\mathcal{M}, \mathcal{N}) =_{\text{df}} \sum_{\substack{m_{\mathbf{O}}: \mathbf{O}^{\mathcal{M}} \rightarrow \mathbf{O}^{\mathcal{N}} \\ m_{\mathbf{A}}: \prod_{\mathbf{x}: \mathbf{O}^{\mathcal{M}}} \mathbf{A}^{\mathcal{M}}(\mathbf{x}) \rightarrow \mathbf{A}^{\mathcal{N}}(m_{\mathbf{O}}(\mathbf{x}))}} \prod_{\substack{\mathbf{x}: \mathbf{O}^{\mathcal{M}} \\ \mathbf{y}: \mathbf{A}^{\mathcal{M}}(\mathbf{y})}} \mathbf{R}^{\mathcal{M}}(\mathbf{y}) \rightarrow \mathbf{R}^{\mathcal{N}}(m_{\mathbf{O}}(\mathbf{x})(\mathbf{y}))$$

As suggested by the notation above, for any $m: \text{Hom}_{\mathcal{L}}(\mathcal{M}, \mathcal{N})$ and $K \in \text{ob}\mathcal{L}$ we will write $m_K(\mathbf{x})$ for the component of m corresponding to sort K in some choice of variables \mathbf{x} .

Definition 4.3 (Split Surjective Map). Let $f: A \rightarrow B$ be any map. Then f is a **split surjection** if the following type is inhabited

$$\mathbf{sps}(f) =_{\text{df}} \prod_{y: B} \sum_{x: A} f(x) = y$$

Remark 4.4. The type $\mathbf{sps}(f)$ is not a proposition, and this is necessary. We must therefore think of $\mathbf{sps}(f)$ as the type of *splittings* of f and there could be many distinct such splittings.

Definition 4.5. Let $m: \text{Hom}_{\mathcal{L}}(\mathcal{M}, \mathcal{N})$. We define

$$\mathbf{fibsurg}(m) =_{\text{df}} \mathbf{sps}(m_{\mathbf{O}}) \times \prod_{\mathbf{x}: \mathbf{O}} \mathbf{sps}(m_{\mathbf{A}}(\mathbf{x})) \times \prod_{\substack{\mathbf{x}: \mathbf{O} \\ \mathbf{y}: \mathbf{A}(\mathbf{x})}} \mathbf{sps}(m_{\mathbf{R}}(\mathbf{x}, \mathbf{y}))$$

We define the type of **fiberwise surjective maps** from \mathcal{M} to \mathcal{N} as

$$\mathbf{FibSurj}(\mathcal{M}, \mathcal{N}) =_{\text{df}} \sum_{m: \text{Hom}_{\mathcal{L}}(\mathcal{M}, \mathcal{N})} \mathbf{fibsurg}(m)$$

We call a term $m: \mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$ a **fiberwise surjection**.

Remark 4.6. Once again, we need to make clear here that $\mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$ is not a Σ -type comprised of an \mathcal{L} -homomorphism that satisfies a property, since $\mathbf{fibsurg}(f)$ will not in general be a proposition. One must rather think of fiberwise surjective maps as \mathcal{L} -homomorphisms together with a *choice* of a splitting that witnesses the fact that it is surjective.

Notation. As usual we will abuse notation and treat $m: \mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$ as if it were comprised purely of its “functional” part. We will thus describe the action of m on terms of \mathcal{M} simply by appending m to the given terms, suppressing subscripts. For example, we will write mx for the result of applying $m_{\mathbf{O}}$ to some $x: \mathbf{O}^{\mathcal{M}}$ and similarly we will write mf for the result of applying $m_{\mathbf{A}}(\mathbf{x})$ to some $f: \mathbf{A}^{\mathcal{M}}(\mathbf{x})$. When it is required to make the particular splitting with which some $m: \mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$ comes equipped we will refer to it by s and describe its action in the same abbreviated manner as we do with m .

Definition 4.7 (Type-Theoretic FOLDS Equivalence). For any FOLDS signature \mathcal{L} (of height 3) and any $\mathcal{M}, \mathcal{N}: \mathbf{Str}_{\mathcal{L}}$ we define

$$\mathcal{M} \simeq_{\mathcal{L}} \mathcal{N} =_{\text{df}} \sum_{\mathcal{P}: \mathbf{Str}_{\mathcal{L}}} \mathbf{FibSurj}(\mathcal{P}, \mathcal{M}) \times \mathbf{FibSurj}(\mathcal{P}, \mathcal{N})$$

Exactly analogously we define $\simeq_{\mathcal{L}}$ on $\mathbf{Mod}_{\mathbb{T}}$, $\mathbf{SatStr}_{\mathcal{L}}$ and \mathbf{SatMod}_T for any \mathcal{L} -theory \mathbb{T} .

Definition 4.7 thus gives us a relation $\simeq_{\mathcal{L}}: \mathbf{SatMod}_{\mathbb{T}} \times \mathbf{SatMod}_{\mathbb{T}} \rightarrow \mathcal{U}$. All the components are now in place for us to prove our higher structure identity principle. To do so, we first require the following lemma.

Lemma 4.8. *Let \mathcal{L} be a FOLDS signature with $h(\mathcal{L}) = 3$, \mathcal{M} and \mathcal{N} be saturated \mathcal{L} -structures, and $m: \mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$ a fiberwise surjective \mathcal{L} -homomorphism.*

- (1) *For any $f: A(\mathbf{x}), g: A(\mathbf{y})$ we have $\text{Ind}^{\mathcal{N}}(mf, mg) \leftrightarrow \text{Ind}^{\mathcal{M}}(f, g)$. In particular, we have $mf \cong mg \leftrightarrow f \cong g$.*
- (2) *For any $x, y: O$ we have $(mx \cong^{\mathcal{M}} my) \simeq (x \cong^{\mathcal{N}} y)$.*

Proof. For (1) we have that $\text{Ind}^{\mathcal{M}}(f, g)$ will be of the following form

$$\prod_{\mathbf{x}: \mathbf{O}^{\mathcal{M}}} \mathbf{R}^{\mathcal{M}}(\mathbf{x}, \mathbf{f}) \leftrightarrow \mathbf{R}^{\mathcal{M}}(\mathbf{y}, \mathbf{g})$$

and similarly $\text{Ind}^{\mathcal{N}}(mf, mg)$ will be of the following form

$$\prod_{\mathbf{x}: \mathbf{O}^{\mathcal{N}}} \mathbf{R}^{\mathcal{N}}(\mathbf{x}, m\mathbf{f}) \leftrightarrow \mathbf{R}^{\mathcal{N}}(\mathbf{y}, m\mathbf{g})$$

Since each \mathbf{R} in the above types will be propositions so will the whole of $\text{Ind}^{\mathcal{M}}(f, g)$ and $\text{Ind}^{\mathcal{N}}(mf, mg)$. Therefore, it suffices to prove that they are logically equivalent. To that end define the maps

$$\left(\prod_{\mathbf{x}: \mathbf{O}^{\mathcal{M}}} \mathbf{R}^{\mathcal{M}}(\mathbf{x}, \mathbf{f}) \leftrightarrow \mathbf{R}^{\mathcal{M}}(\mathbf{y}, \mathbf{g}) \right) \xrightarrow{\sigma} \left(\prod_{\mathbf{x}: \mathbf{O}^{\mathcal{N}}} \mathbf{R}^{\mathcal{N}}(\mathbf{x}, m\mathbf{f}) \leftrightarrow \mathbf{R}^{\mathcal{N}}(\mathbf{y}, m\mathbf{g}) \right)$$

$$\eta \longmapsto \lambda \mathbf{x}. (\lambda \mathbf{r}. m(p_1(\eta_s(\mathbf{x}))(s\mathbf{r}), \lambda \mathbf{r}. m(p_2(\eta_s(\mathbf{x}))(s\mathbf{r}))$$

and

$$\left(\prod_{\mathbf{x}: \mathbf{O}^{\mathcal{M}}} \mathbf{R}^{\mathcal{M}}(\mathbf{x}, \mathbf{f}) \leftrightarrow \mathbf{R}^{\mathcal{M}}(\mathbf{y}, \mathbf{g}) \right) \xleftarrow{\tau} \left(\prod_{\mathbf{x}: \mathbf{O}^{\mathcal{N}}} \mathbf{R}^{\mathcal{N}}(\mathbf{x}, m\mathbf{f}) \leftrightarrow \mathbf{R}^{\mathcal{N}}(\mathbf{y}, m\mathbf{g}) \right)$$

$$\lambda \mathbf{x}. (\lambda \mathbf{r}. s(p_1(\epsilon_{\mathbf{x}})(m\mathbf{r})), \lambda \mathbf{r}. s(p_2(\epsilon_{\mathbf{x}})(m\mathbf{r}))) \longmapsto \epsilon$$

The fact that $mf \cong mg \leftrightarrow f \cong g$ follows by as the special case when $\mathbf{x} = \mathbf{y}$.

For (2), we define maps as shown:

$$\begin{array}{ccc}
 x \cong y & \xrightleftharpoons[e]{d} & mx \cong my \\
 \parallel & & \parallel \\
 \prod_{\mathbf{z}: \mathbf{O}^{\mathcal{M}}} \mathbf{A}^{\mathcal{M}}(x, \mathbf{z}) \simeq \mathbf{A}^{\mathcal{M}}(y, \mathbf{z}) & \xrightleftharpoons[e]{d} & \prod_{\mathbf{z}: \mathbf{O}^{\mathcal{N}}} \mathbf{A}^{\mathcal{N}}(mx, \mathbf{z}) \simeq \mathbf{A}^{\mathcal{N}}(my, \mathbf{z}) \\
 \parallel & & \parallel \\
 \prod_{\substack{\mathbf{z}: \mathbf{O}^{\mathcal{M}} \\ f: A^{\mathcal{M}}(x, \mathbf{z})}} \sum_{g: A^{\mathcal{M}}(y, \mathbf{z})} \text{Ind}^{\mathcal{M}}(f, g) & \xrightleftharpoons[e]{d} & \prod_{\substack{\mathbf{z}: \mathbf{O}^{\mathcal{N}} \\ f: A^{\mathcal{N}}(mx, \mathbf{z})}} \sum_{g: A^{\mathcal{N}}(my, \mathbf{z})} \text{Ind}^{\mathcal{N}}(mf, mg)
 \end{array}$$

$$\eta \longmapsto \lambda \mathbf{z} \lambda f. \left\langle m(p_1(\eta_{s\mathbf{z}, sf})), \sigma(p_2(\eta_{s\mathbf{z}, sf})) \right\rangle$$

$$\lambda \mathbf{z} \lambda f. \left\langle s(p_1(\epsilon_{m\mathbf{z}, mf})), \tau(p_2(\epsilon_{m\mathbf{z}, mf})) \right\rangle \longleftarrow \epsilon$$

It is easy to check that d and e are inverses to each other, thus establishing the required equivalence. \square

We are now ready to state and prove the main result of this paper which is a fully precise version of the guiding Pre-Theorem 0.2 in the Introduction.

Theorem 4.9 (Higher Structure Identity Principle). *For any \mathcal{L} with $h(\mathcal{L}) = 3$, any \mathcal{L} -theory \mathbb{T} and for any saturated \mathbb{T} -models \mathcal{M} and \mathcal{N} of an \mathcal{L} -theory \mathbb{T} we have*

$$(\mathcal{M} \simeq_{\mathcal{L}} \mathcal{N}) \simeq (\mathcal{M} =_{\mathbf{SatMod}_{\mathbb{T}}} \mathcal{N})$$

Proof. Let \mathcal{M} and \mathcal{N} be saturated models for some \mathcal{L} -theory \mathbb{T} . Clearly, it suffices to prove that $\mathbf{FibSurj}(\mathcal{M}, \mathcal{N}) \simeq \mathcal{M} =_{\mathbf{SatMod}_{\mathbb{T}}} \mathcal{N}$. We define a map

$$k: \mathbf{FibSurj}(\mathcal{M}, \mathcal{N}) \rightarrow \mathcal{M} =_{\mathbf{SatMod}_{\mathbb{T}}} \mathcal{N}$$

Let $m: \mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$. By univalence, to produce a term of type $\mathcal{M} =_{\mathbf{SatMod}_{\mathbb{T}}} \mathcal{N}$ it suffices to produce component-wise equivalences for all the defining data of \mathcal{M} and \mathcal{N} . Since m is assumed fiberwise surjective we know that it provides component-wise surjections between the defining data of \mathcal{M} and \mathcal{N} . Thus, it suffices to prove that m is also component-wise an injection. We do so by taking the data of \mathcal{L} level-by-level, abusing notation by ignoring the necessary transfers.

Since \mathcal{M} and \mathcal{N} are saturated, by Proposition 3.4 each $\mathbf{R}^{\mathcal{M}}(\mathbf{x})$ and $\mathbf{R}^{\mathcal{N}}(\mathbf{y})$ will be of h -level 1. Therefore, all the

$$m_{\mathbf{R}}(\mathbf{x}): \mathbf{R}^{\mathcal{M}}(\mathbf{x}) \rightarrow \mathbf{R}^{\mathcal{N}}(\mathbf{y})$$

are trivially also injections, and therefore equivalences.

Now consider $m_{\mathbf{A}}(\mathbf{x}): A^{\mathcal{M}}(\mathbf{x}) \rightarrow A^{\mathcal{N}}(m\mathbf{x})$. By Proposition 3.4 once again we know that $m_{\mathbf{A}}(\mathbf{x})$ is a map of h -sets and therefore to show that it is an equivalence it suffices to show $m_{\mathbf{A}}(\mathbf{x})(\alpha) = m_{\mathbf{A}}(\mathbf{x})(\beta) \rightarrow \alpha = \beta$. But then note that we have the following composite map

$$m_{\mathbf{A}}(\mathbf{x})(\alpha) = m_{\mathbf{A}}(\mathbf{x})(\beta) \xrightarrow{(1)} m_{\mathbf{A}}(\mathbf{x})(\alpha) \cong m_{\mathbf{A}}(\mathbf{x})(\beta) \xrightarrow{(2)} \alpha \cong \beta \xrightarrow{(3)} \alpha = \beta$$

where (1) is by induction on identity, (2) is by the first part of Lemma 4.8 since m is assumed fiberwise surjective and (3) is because \mathcal{M} is assumed saturated.

Finally, for $m_{\mathbf{O}}: \mathbf{O}^{\mathcal{M}} \rightarrow \mathbf{O}^{\mathcal{N}}$, to show that it is injective it suffices to show that there is an equivalence $(x = y) \simeq (mx = my)$ for any $x, y: \mathbf{O}$. Since both \mathcal{M} and \mathcal{N} are saturated this is equivalent to showing that $(x \cong y) \simeq (mx \cong my)$. But since m is fiberwise surjective, this follows from the second part of Lemma 4.8.

This completes the definition of the map k . Now we need to show that k is itself an equivalence. To see this note that by univalence any $p: \mathcal{M} =_{\mathbf{SatMod}_{\mathbb{T}}} \mathcal{N}$ gives rise to a fiberwise surjective homomorphism m_p with components given by the equivalences that p induces. This gives a map

$$l: \mathcal{M} =_{\mathbf{SatMod}_{\mathbb{T}}} \mathcal{N} \rightarrow \mathbf{FibSurj}(\mathcal{M}, \mathcal{N})$$

which can easily be seen to be a two-sided inverse for k . Thus, k is an equivalence, and this completes the proof. \square

In order to test the adequacy of our Structure Identity Principle we need to show that it specializes to familiar cases. In this case, we need to show that Theorem 9.4.16 of [Uni13] follows as a special case of our Theorem 4.9. The next series of Propositions and Lemmas aim exactly at demonstrating this.

Proposition 4.10. *The type $\mathbf{SatMod}_{\mathbb{T}_{cat}}^2$ of 2-saturated models of the \mathcal{L}_{cat} -theory of categories is equivalent to the type \mathbf{PreCat} of precategories.*

Proof. By the axioms for $=_A$ in the theory of categories we can obtain that

$$\mathbb{T}_{cat} \models \forall x, y: O \forall f, g: A(x, y). f \cong g \leftrightarrow f =_A g$$

So if \mathcal{C} is any 2-saturated \mathbb{T}_{cat} -model we will have

$$\prod_{x, y: O^{\mathcal{C}}} \prod_{f, g: A^{\mathcal{C}}(x, y)} (f =_A g)^{\mathcal{C}} \leftrightarrow f =_{A^{\mathcal{C}}(x, y)} g$$

And since $(f =_A g)^{\mathcal{C}}$ will be of h -level 1 (by Proposition 3.4) this means exactly that $A^{\mathcal{C}}(x, y)$ is a set for all $x, y: O^{\mathcal{C}}$. What remains to be shown is that the (interpretations of) the axioms of \mathbb{T}_{cat} correspond exactly to the axioms for precategories (but in relational form). This is straightforward and already established in [Tse16] as Proposition 7.1 and has even been formalized in [VAG⁺]. \square

Lemma 4.11. *Let \mathcal{C} be a precategory. For every $x, y, z: O^{\mathcal{C}}$, $\alpha: A^{\mathcal{C}}(z, x) \rightarrow A^{\mathcal{C}}(z, y)$ and $h: A^{\mathcal{C}}(z, x)$ we have*

$$\mathrm{Ind}^{\mathcal{C}}(h, \alpha(h)) \simeq \prod_{\substack{w: O^{\mathcal{C}} \\ g: A^{\mathcal{C}}(w, z)}} \alpha(h) \circ g = \alpha(h \circ g)$$

Proof. Since both types are propositions (since we assume that \mathcal{C} is a precategory, and therefore identity in any hom-set will be a proposition) it suffices to show that they are logically equivalent.

For

$$\text{Ind}^{\mathcal{C}}(h, \alpha(h)) \rightarrow \prod_{\substack{w: O^{\mathcal{C}} \\ g: A^{\mathcal{C}}(w, z)}} \alpha(h) \circ g = \alpha(h \circ g)$$

note that by the definition of Ind , $\text{Ind}^{\mathcal{C}}(h, \alpha(h))$ will contain the following proposition among the cartesian product that defines it:

$$\prod_{\substack{w: O \\ g: A(w, z)}} (\alpha(h) \circ g = \alpha(h \circ g) \leftrightarrow h \circ g = h \circ g)$$

Since the RHS is trivially inhabited, the above type logically entails

$$\prod_{\substack{w: O^{\mathcal{C}} \\ g: A^{\mathcal{C}}(w, z)}} \alpha(h) \circ g = \alpha(h \circ g)$$

as required.

In the other direction, the only non trivial biconditional in $\text{Ind}^{\mathcal{C}}(h, \alpha(h))$ given the data will be

$$\prod_{\substack{w, w': O^{\mathcal{C}} \\ g: A^{\mathcal{C}}(w, z) \\ k: A^{\mathcal{C}}(x, w')}} (g \circ h = k \leftrightarrow g \circ \alpha(h) = \alpha(k))$$

But this follows immediately from

$$\prod_{\substack{w: O^{\mathcal{C}} \\ g: A^{\mathcal{C}}(w, z)}} \alpha(h) \circ g = \alpha(h \circ g)$$

as required. \square

Lemma 4.12. *Let \mathcal{C} be a precategory. For every $x, y: O^{\mathcal{C}}$ we have*

$$\prod_{z: O^{\mathcal{C}}} A(z, x) \simeq^{\mathcal{C}} A(z, y)$$

Proof. By Lemma 3.5 we know that $A(z, x) \simeq^{\mathcal{C}} A(z, y)$ is equivalent to the type of bijections from $A^{\mathcal{C}}(z, x)$ to $A^{\mathcal{C}}(z, y)$. Since each \mathcal{C} is a precategory, $A^{\mathcal{C}}(z, x)$ and $A^{\mathcal{C}}(z, y)$ are sets, and therefore bijections are equivalences, which means that $A(z, x) \simeq^{\mathcal{C}} A(z, y)$ is the type of (semantic) equivalences $A^{\mathcal{C}}(z, x) \simeq A^{\mathcal{C}}(z, y)$. Exactly the same reasoning applies to $A(x, z) \simeq^{\mathcal{C}} A(y, z)$, $A(x, x) \simeq^{\mathcal{C}} A(x, y)$, $A(x, x) \simeq^{\mathcal{C}} A(y, y)$ and $A(x, y) \simeq^{\mathcal{C}} A(y, y)$ which are the rest of the conjuncts of $x \cong y$ if we follow Definition 3.1. Thus, to prove the Lemma it suffices to show

$$\left[\prod_{z: O^{\mathcal{C}}} A^{\mathcal{C}}(z, x) \simeq A^{\mathcal{C}}(z, y) \right] \simeq \left[\prod_{z: O^{\mathcal{C}}} (A^{\mathcal{C}}(z, x) \simeq A^{\mathcal{C}}(z, y)) \times (A^{\mathcal{C}}(x, z) \simeq A^{\mathcal{C}}(y, z)) \right]$$

which follows from the elementary result in category theory that precomposition by an arrow induces a bijection on every hom-set iff postcomposition also does. \square

Proposition 4.13. *The type $\text{SatMod}_{\mathbb{T}_{\text{cat}}}$ of totally saturated models of the \mathcal{L}_{cat} -theory of categories is equivalent to the type UniCat of univalent categories.*

Proof. By Proposition 4.10 it suffices to show that for any precategory \mathcal{C} and any $x, y: O^{\mathcal{C}}$ we have $\text{Iso}^{\mathcal{C}}(x, y) \simeq x \cong y$. To see this observe the following series of equivalences:

$$\begin{aligned}
& \text{Iso}^{\mathcal{C}}(x, y) \\
& \downarrow \text{By Definition} \\
& \sum_{f: A^{\mathcal{C}}(x, y)} \text{isiso}(f) \\
& \downarrow \text{By Yoneda Lemma} \simeq \\
& \prod_{z: O^{\mathcal{C}}} \prod_{\alpha: A^{\mathcal{C}}(z, x) \rightarrow A^{\mathcal{C}}(z, y)} \sum \text{bij}(\alpha) \times \prod_{\substack{w: O^{\mathcal{C}} \\ h: A^{\mathcal{C}}(z, x) \\ g: A^{\mathcal{C}}(w, z)}} \alpha(h) \circ g = \alpha(h \circ g) \\
& \downarrow \text{By Lemma 4.11} \simeq \\
& \prod_{z: O^{\mathcal{C}}} \prod_{\alpha: A^{\mathcal{C}}(z, x) \rightarrow A^{\mathcal{C}}(z, y)} \sum \text{bij}(\alpha) \times \prod_{h: A^{\mathcal{C}}(z, x)} \text{Ind}^{\mathcal{C}}(h, \alpha(h)) \\
& \downarrow \text{By Lemma 3.5} \simeq \\
& \prod_{z: O^{\mathcal{C}}} \left[\prod_{f: A(z, x)g: A(z, y)} \sum \text{Ind}(f, g) \times \prod_{h: A(z, y)} \text{Ind}(f, h) \rightarrow h = g \right] \\
& \quad \times \\
& \left[\prod_{\substack{f, f': A^{\mathcal{C}}(z, x) \\ g, g': A^{\mathcal{C}}(z, y)}} \text{Ind}^{\mathcal{C}}(f, g) \times \text{Ind}^{\mathcal{C}}(f', g') \times g = g' \rightarrow f = f' \right] \\
& \quad \times \\
& \left[\prod_{g: A^{\mathcal{C}}(z, x)f: A^{\mathcal{C}}(z, y)} \sum \text{Ind}^{\mathcal{C}}(f, g) \right] \\
& \quad \parallel \\
& \prod_{z: O^{\mathcal{C}}} A(z, x) \simeq^{\mathcal{C}} A(z, y) \\
& \downarrow \text{By Lemma 4.12} \simeq \\
& x \cong y
\end{aligned}$$

□

Corollary 4.14 ([Uni13], Theorem 9.4.16). *For any univalent categories \mathcal{C} and \mathcal{D} , the type of categorical equivalences $\mathcal{C} \simeq_{\text{cat}} \mathcal{D}$ is equivalent to $\mathcal{C} =_{\text{UniCat}} \mathcal{D}$.*

Proof. By Theorem 4.9 what remains now is to show that in **UniCat**, the relation $\simeq_{\mathcal{L}_{\text{cat}}}$ is equivalent to categorical equivalence as defined in [Uni13]. But this is immediate by inspection. \square

Remark 4.15. Propositions 4.10 and 4.13 in our opinion provide the correct taxonomy of category theory in the Univalent Foundations. In particular, we should regard pre-categories not as “unsaturated” categories but rather as the 2-saturated \mathbb{T}_{cat} -models. This leaves two more “category theories” that currently have no name: the “totally unsaturated” models $\mathbf{Mod}_{\mathbb{T}_{\text{cat}}}$ and the 1-saturated models $\mathbf{SatMod}_{\mathbb{T}_{\text{cat}}}^1$. There are thus four different category theories in the Univalent Foundations with respect to saturation as we have defined it. They can be thought of as follows, with the obvious inclusions:

$$\mathbf{UniCat} \hookrightarrow \mathbf{PreCat} \hookrightarrow \mathbf{SatMod}_{\mathbb{T}_{\text{cat}}}^1 \hookrightarrow \mathbf{Mod}_{\mathbb{T}_{\text{cat}}}$$

We believe this is the right point of view to take with respect to category theory (and other theories) in the Univalent Foundations.

Remark 4.16. As the series of inclusions above suggests, clearly one can now ask to what extent we can have “Rezk completion”-type operations that reverse (in appropriate ways) the above inclusions. This is clearly an avenue worth pursuing further. In particular, the following question seems to us of immediate interest: Given an \mathcal{L} -theory \mathbb{T} , for which $n \geq h(\mathcal{L})$ can we construct an adjoint to the inclusion $\mathbf{SatMod}_{\mathbb{T}_{\text{cat}}}^n \hookrightarrow \mathbf{SatMod}_{\mathbb{T}_{\text{cat}}}^{n-1}$? Answers to this question seem to us unlikely to be pursued at least initially for any \mathcal{L} with $h(\mathcal{L}) > 3$.

Finally, let us mention that in principle there is no obstacle to extending the proof of Theorem 4.9 to signatures \mathcal{L} of arbitrarily large finite height. There are however two main difficulties. Firstly, finding a sufficiently practical way of packaging the syntax of such an \mathcal{L} so as to define the relevant notions of \mathcal{L} -homomorphism, \mathcal{L} -equivalence etc. Secondly, proving the higher analogues of Lemma 4.8 which would require establishing equivalences between n -types of ever larger n . Solving these difficulties seems not entirely out of reach. Proving a version of Theorem 4.9 for all $n > 1$ therefore seems to us an interesting open problem.

Acknowledgments. I thank Peter LeFanu Lumsdaine for the discussion that initiated this paper. This work was partially supported by NSF DMS-1554092 (P.I. Harry Crane).

REFERENCES

- [Mak95] M. Makkai, *First order logic with dependent sorts with applications to category theory*, 1995. <http://www.math.mcgill.ca/makkai/folds/foldsinpdf/FOLDS.pdf>.
- [Tse16] D. Tsementzis, *Homotopy Model Theory I: Syntax and Semantics*, arXiv preprint arXiv:1603.03092 (2016).
- [Tse] ———, *Homotopy Model Theory II: Completeness for 1-Logic*. In Preparation.
- [Uni13] Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics* (2013). <http://homotopytypetheory.org/book>.
- [VAG⁺] V. Voevodsky, B. Ahrens, D. Grayson, et al., *UniMath: Univalent Mathematics*. Available at <https://github.com/UniMath>.

DEPARTMENT OF PHILOSOPHY, PRINCETON UNIVERSITY

Current address: Princeton, NJ 08544, USA

E-mail address: dtsement@princeton.edu

DEPARTMENT OF STATISTICS AND BIOSTATISTICS, RUTGERS UNIVERSITY

Current address: New Brunswick, NJ 08854, USA

E-mail address: dt506@rci.rutgers.edu